

Digital Business Ecosystem

Contract n° 507953

Workpackage 18

**Socio-economic perspectives of sustainability
and dynamic specification of behaviour
in Digital Business Ecosystems**

Deliverable 18.6

**5-Year living roadmap
for Digital Ecosystems research
in biology-inspired computing**

Contract Number: 507953

Project Acronym: DBE

Title: Digital Business Ecosystem

Deliverable N°: 18.6

Due date: 31/12/2006

Delivery Date : 13/03/2007

Short Description: This report presents a 5-year roadmap for bio-inspired computing in the area of Digital Ecosystems research.

Partners owning: Paolo Dini (LSE)

Partners contributed: Gerard Briscoe (HWU)

Made available to: unlimited distribution

Versioning

Version	Date	Author, Organisation
1	11 March 07	Paolo Dini, LSE
2	13 March 07	Paolo Dini, LSE

Quality check

1st Internal Reviewer: Miguel Vidal (SUN)

2nd Internal Reviewer: Paul Krause (UNIS)



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License. To view a copy of this license, visit : <http://creativecommons.org/licenses/by-nc-sa/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, US

Table of Contents

1. Introduction	3
2. Theoretical Framework for Biology-Inspired Computing	4
3. Toward a Roadmap for Biocomputing	6
4. Conclusion	7

1. INTRODUCTION

The work performed over the past 3 years has attempted to arrive at an operational understanding of emergent computation, by which we mean useful behaviour that has not been pre-programmed in advance. Usefulness, however, is not an absolute notion and can only be defined relative to a particular context. One way to make computation relevant (even if not necessarily useful) to a particular context is to make that context an integral part of the computation. We describe this scenario as computation through a distributed algorithm.

We can envision context-dependent computation as an extreme form of the concept of reuse in software engineering, but where the reuse happens dynamically at run-time. The details of how the modules outside a particular “system” participate in that system’s computation are not as important as the fact that any form of participation requires communication, and communication implies some form of interaction. This is how we came to the realisation that emergent computation requires both the concept of distributed algorithm and interaction.

An additional requirement appears to be that interactive computation needs to be symbiotic, meaning that the same computation in which two or more systems participate through their interaction serves a different purpose and has a different meaning (semantics) for each system. It is in effect overloaded, and can be seen at an abstract level as analogous to a set of unknowns that simultaneously satisfies more than one equation (or constraint).

We can further distinguish between two possibilities for interaction dynamics: synchronous and asynchronous. The former is associated with the concept of dynamical coherence in physics and with reactive or real-time systems in computer science. The latter is associated with dynamical incoherence in physics and with memory-based systems, or agents, in computer science. We believe that this distinction, although obviously useful in both theoretical and practical discussions, is in fact inconsequential here. In other words, computational emergence can apply to both types of system interactions, although a presence of memory would seem to indicate the ability to support more complex behaviour.

Although the discussion so far has only made references to software engineering, computer science, physics, and mathematical concepts, the picture we are drawing is remarkably close to our current understanding of several of the principles according to which biological systems operate. It therefore seems advantageous to integrate biology explicitly into the research methodology. The principles in question underpin the set of cell biochemical reaction pathways that can be loosely grouped under the umbrella term of “gene expression” and that are based on interactions, in an abstract sense, and specifically on chemical reactions. This is the bias taken in this short report.

The 5-year roadmap presented here embodies the output of the past 3.5 years of research in the area of emergent computation for digital ecosystems. This output is more in the form of high-level connections, analogues, and associations between sometimes vastly different disciplines than in the form of usable algorithms or architectures. The picture we have arrived at indicates the need to take into account different aspects of a large number of sub-disciplines of mathematics, computer science, biology, physics, language, and engineering in order to arrive at an integrated model that can be translated into useful algorithms and architectures. In this short report we aim to show a distillation of the disciplines involved, and a potentially fruitful route to arrive at the realisation of biology-inspired computing.

2. THEORETICAL FRAMEWORK FOR BIOLOGY-INSPIRED COMPUTING

Biology-inspired computing (“biocomputing” for short) is a rapidly growing sub-discipline of computer science that is diversifying into several areas of potential interest to Digital Ecosystems research. In addition to the older genetic algorithms and neural networks, which originated in the 1950s, artificial life has become popular in the early 1990s along with cellular automata, swarms, immune system-inspired computing, and others. More recently, computational systems biology has emerged as a very interesting field where biologists and computer scientists work together on *biology* problems through computer science models.¹ In this short deliverable we are not attempting to provide a review of biology-inspired computing. Rather, we wish to develop a roadmap for computing that is biased toward gene expression.

Without taking anything away from the other branches of biocomputing, we contend that gene expression is of greater short-term relevance to the DBE Evolutionary Environment because it attempts to reconcile the short-term morphogenetic and metabolic functions of DNA with its hereditary function in a self-optimising natural selection dynamics. We believe that the former are of central importance for the development of 4th and 5th-generation programming languages, which are envisioned to enable the coupling of semantic specification with context-sensitive syntactical implementation and run-time instantiation. The latter is regarded as an essential feature of future large-scale, distributed, complex software systems.

The motivation for our vision of gene expression-inspired computing relies on a relatively simple observation. DNA can be seen as a pattern “written” in a molecular alphabet (of 4 symbols) that is characterised by two apparently contrasting properties:

- 1- Many parts of the DNA molecule recur either in structure or in fact identically in many completely different life forms, suggesting a large level of reuse of what we could call design patterns
- 2- The apparent limitless variation potential of life forms indicates that such common structural features can still support and do not prevent the continuous emergence of an endless variety of organism morphologies

These two properties taken together suggest that the variation in life forms does not interfere with the conservation of the ability to “express” DNA into these life forms. This has engineering significance, because it suggests that whereas the dynamic processes through which the DNA patterns are expressed into their corresponding phenotypes are quite constrained and fairly robust, the DNA patterns themselves appear to be quite arbitrary (beyond the common structures already indicated). This is of course consistent with our understanding of the randomising effects of the genetic operators. The significance of this observation is that such arbitrary patterns are compatible with the similarly arbitrary process by which software engineers specify the structural and functional characteristics of software artefacts. Thus if our understanding of gene expression is correct it appears to incorporate the solution to finding the difficult balance between general laws and context dependence. It appears to be able to strengthen the engineering process by introducing additional structure in the process itself, without however constraining in any way the freedom to express any requirements at user level.

The domain modelling and requirements gathering phase of software engineering (i.e. UML, MDA, etc) is called “arbitrary” because it has undergone a continuous process of abstraction since its inception precisely to reach total independence from the underlying implementation and technologies. At the same time, however, engineering is also concerned with the non-functional requirements, the technical architecture, the structural architecture, and so forth. It is not clear to what extent the bio-inspired approach can replace or even simply augment these aspects of software engineering. Our contention is to focus on proving that it can help with even only one of them, and postpone discussions of extent of scope until later.

¹ In addition to the dichotomy between biocomputing and computational biology as defined here, there is also the field of bioinformatics, which is however not of central interest to this research since it is mainly concerned with the support of biological calculations and data structures by computing means, rather than with developing fundamentally new ways of thinking about computing.

One of the main objectives of biocomputing, therefore, is to arrive at a specification language that can mirror these desirable properties of DNA, since such a language would appear not to interfere with existing, and continually evolving, software engineering and domain modelling languages. These so-called 4th and 5th-generation languages are already approaching the ability to compile specifications expressed in natural language into running code automatically, but their awareness of the context remains limited. Assuming that this general trend continues, the gene expression approach at biocomputing acquires greater significance once we realise that the “software DNA” could be interfaced directly to human languages. The significance here is not so much functionalist, although business model specification in natural language is certainly appealing, as epistemological. In fact, if we accept the view that language is a social process, this framework reinforces the interpretation of the development of technology as the expression of a cultural process, which however relies on the structuralist, functionalist, and biology-inspired rules to optimise the link between language-based specification and functional behaviour.

This appears to offer a theoretical explanation for the productive coexistence of the intersubjective epistemology of language and social processes with the objectivist epistemology of mathematics and of bi-modal logic that underpins computer science. Whereas this link has been at the core of the interpretation of technology development as cultural production, the introduction of mathematical constraints inspired by biology allows the systematisation of the process, which brings it within the realm of engineering design. Quite remarkably, therefore, a thread of research (WP18) that started from a connectionist and bottom-up perspective is now pointing toward the legitimisation of top-down engineering design through the recognition of its ultimately social nature, and toward the enablement of the automatic generation of code and behaviour through a close mapping of the regularities underpinning biological systems into structural and interaction rules that support distributed symbiotic computation.

Fig. 1 attempts to capture the main theoretical disciplines and some of the more specific models and languages that are believed to be important to this vision of computing. The arrows should be understood as a combination of workflows, knowledge flows, and dependencies (expressed in the opposite sense to the usual convention used in class diagrams).

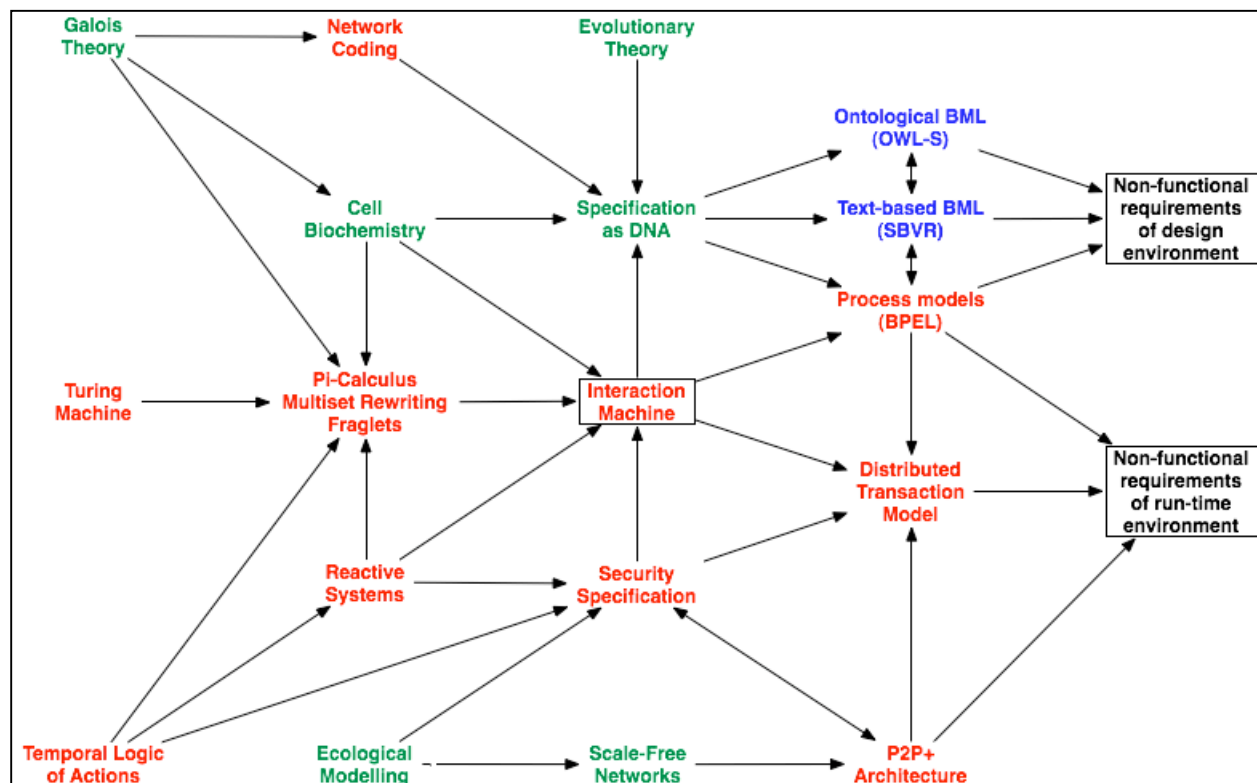


Fig. 1 Ontological workflows of theoretical and applied disciplines underpinning biocomputing

3. TOWARD A ROADMAP FOR BIOCOMPUTING

Fig. 2 begins to translate the disciplinary ontology of Fig. 1 into a sequential workflow in 5 stages that could partially overlap. Stage 1 focuses on the integration of a large number of theories and models (9 are shown in the figure, but more could be added), and operationally could last anywhere from 1-3 years. The outputs of this first stage will be needed by Stage 2 as it attempts to reach three ambitious goals:

- Specification as DNA: a meta-specification language that will define the characteristics that specifications need to have in order to support DNA-like expressive properties within a context of interactive computing
- Interaction Machine: a new model of computation that takes as its point of departure Turing's 1936 paper and, by building on concurrency theory and chemical programming, generalises the results to a wider range of computational machines that are inclusive of interaction with an external environment
- P2P+ Architecture: a self-bootstrapping "pure" peer-to-peer network architecture based on dynamic topologies inspired by scale-free and small-world models.

Reaching these three goals will enable us to derive the non-functional requirements (as discussed in D18.4) on both the design and run-time environments, Stage 3. This will, in turn, enable us to develop new languages or modify existing languages that are currently being used in different aspects of software engineering and networking, in Stage 4. These modifications will be completely transparent to the users. Stage 4 can then be understood as the construction (or modification) of the tools that are used to express (in the usual sense) the functional requirements of the users. Finally, Stage 5 will address the integration of all these architectural requirements into an integrated Evolutionary Environment that will link seamlessly to the social systems and business ecosystems to realise the vision of Digital Ecosystems.

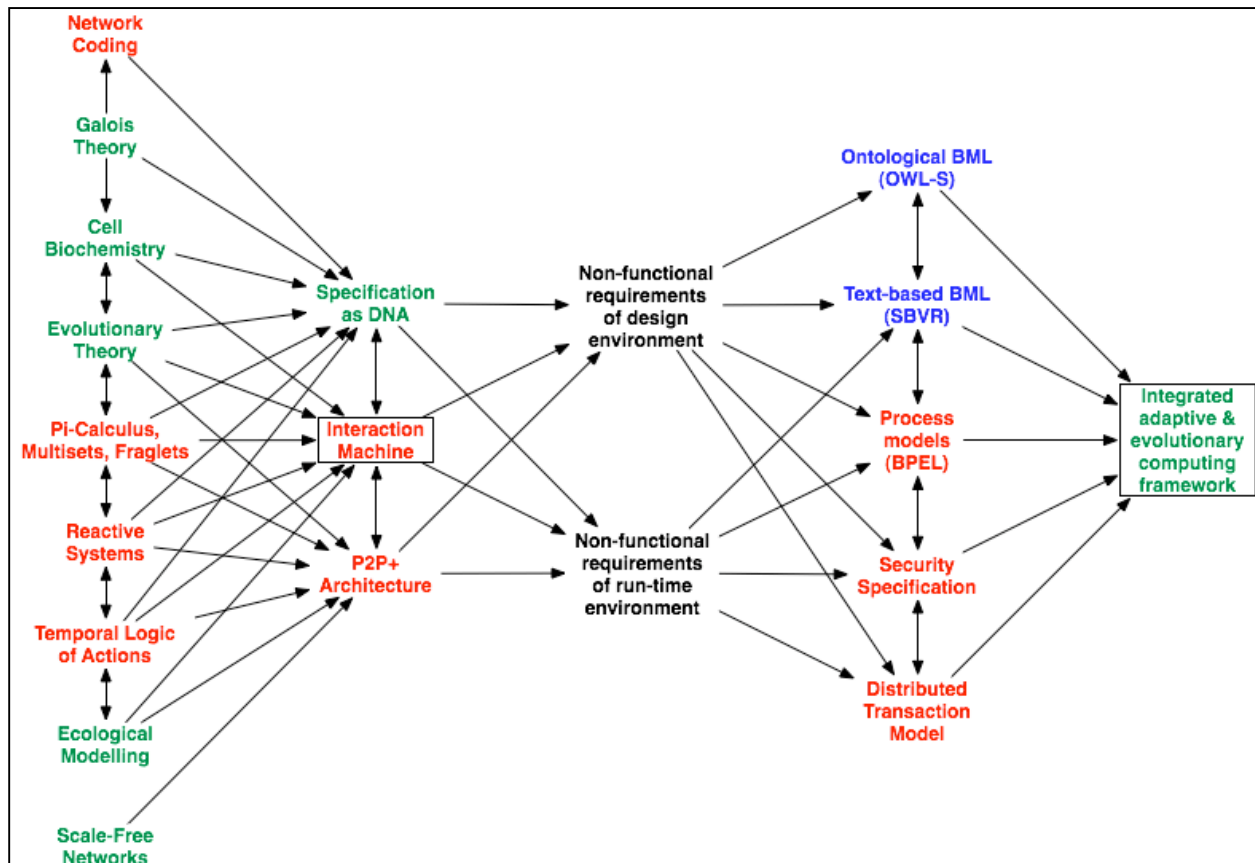


Fig. 2 Time-oriented ontological workflow and dependence diagram for biocomputing

It is a little counter-intuitive to produce a 5-year roadmap that is largely based on the assumption of the validity of a symmetry-based mathematical approach at deriving non-functional requirements for the design and run-time environments. However such a roadmap was requested, so it was produced by

extrapolating the best theoretical results we could discover in the literature or derive ourselves during the course of the DBE project (Fig. 3).

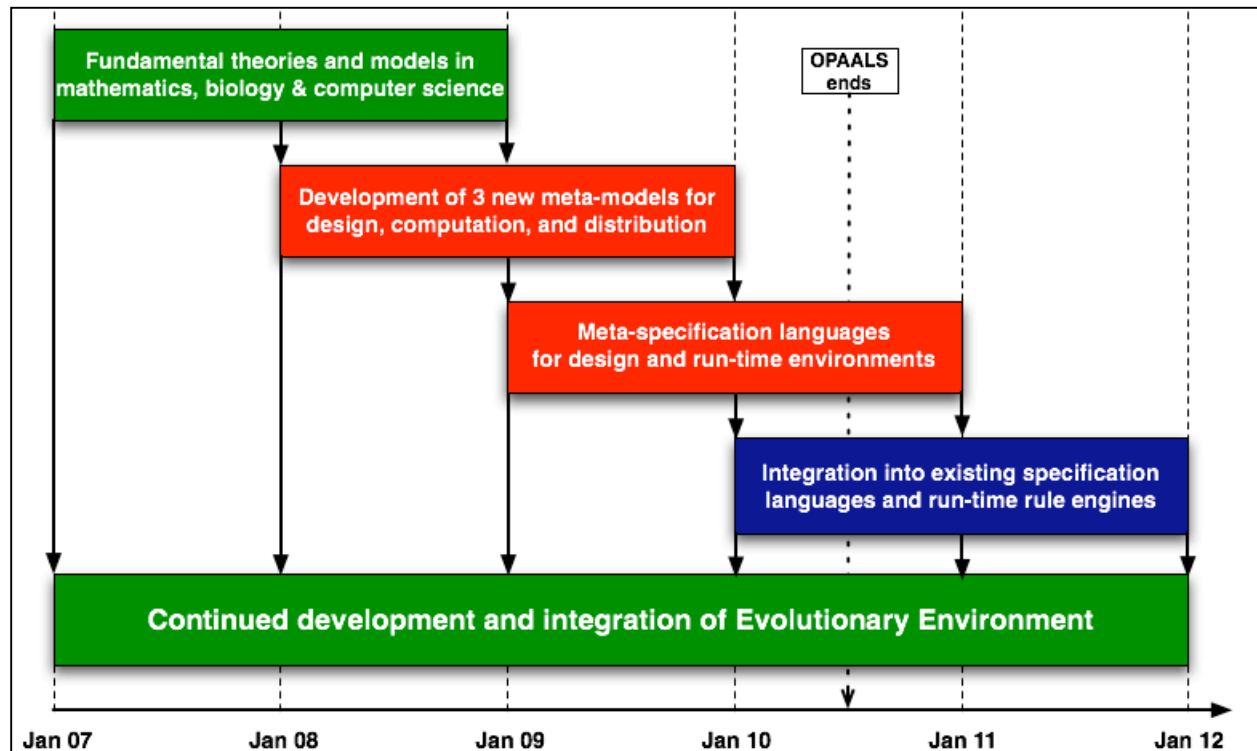


Fig. 3 5-year roadmap of research activities aiming to realise the biocomputing vision

4. CONCLUSION

The roadmap provided should not be regarded as a roadmap for Digital Ecosystems research. It is only a roadmap for the emergence and realisation of the biocomputing vision within the much broader interdisciplinary field of Digital Ecosystems research. Thus, for example, a lot of work needs to happen in business modelling languages before Jan 2010. We need to continue our efforts to understand the processes of formalisation of knowledge and the expression of business requirements into languages that have so far remained fairly abstract and rigid relative to natural language. The socio-economic aspects of Digital Ecosystems research are not addressed at all by this report or by the above roadmap. Such research needs to continue if we want Digital Ecosystems to become sustainable and relevant to a broad array of different social, economic, technological, and cultural contexts.

From the point of view of biocomputing, the 3-year DBE project therefore ends with something of a gamble. The EC requested a contribution from the physical and natural sciences in the development of a new biology-inspired computing paradigm. In view of the challenges posed by fitness evaluation based on business semantics, we discovered that this contribution will need to involve a lot more than genetic algorithms. As different connectionist, neural network-based, and evolutionary methods become increasingly integrated in the EvE, in this report we have highlighted another set of research areas that appear to be necessary in order to integrate also a symbol-based view of computing into the same Evolutionary Environment. To face this challenge we will have to develop a common language of research that transcends disciplinary differences and traditions: we will need to become a community of practice whose members are willing to cooperate and communicate. Only then will we be able to speak of biocomputing as a new paradigm² and, by extension, only in this manner will we be able to achieve the Digital Ecosystems vision.

² Kuhn, T (1996). *The Structure of Scientific Revolutions*, 3rd Ed, University of Chicago Press.